

# agent2d

スタートアップマニュアル

秋葉原プログラミング教室

サッカー部編

2019/4/7

## 目次

## 内容

目次.....	1
1. セットアップ .....	3
2. agent2d の使い方 .....	3
(1) 参考資料.....	3
(2) librcsc(ライブラリ).....	4
(3) フォルダ構造 .....	4
①コンパイル前 .....	4
②コンパイル後 .....	11
(4) コンパイル .....	21
①コンパイル (ビルドの過程) .....	21
②Make.....	21
③GNU Autotool .....	21
④agent2d でファイルを追加した時 .....	22
⑤agent2d のライブラリをプログラムに同梱する方法 .....	22
(5) プログラムの開始までの流れ (start.sh) .....	23
①LIBPATH の書き込み (12 行~20 行) .....	23
②実行ファイル、conf ファイル、port 番号などを変数に入れる (24 行~53 行) ...	23
・ 検討事項 (2019/4/7) .....	23
③ヘルプの内容表示 (55 行~80 行) .....	23
④引数の処理 (82 行~262 行) .....	23
⑤その他の処理 (264 行~) .....	23
⑥start.sh を実行した後にコンソールに表示される内容 .....	23
(6) 修正を必要とするファイル.....	30
①フォーマーションの変更 .....	30
②ボールキックの行動評価関数.....	30
③ポジショニング動作 .....	31
④戦略 (ポジショニング) .....	31
⑤コミュニケーションルール .....	32
⑥プレイヤーの役割.....	32
⑦全体のコントロール .....	32
3. フォーマーションの変更 .....	33
(1) フォルダ構造 .....	33

(2) fedit2 の使い方 .....	34
検討事項 (2019/4/7) .....	34
参考情報 .....	36
(1) Agent2d 関連 .....	36
(2) 世界大会出場チームのアルゴリズム説明 (Team Description Paper) .....	36
(3) C、C++関連 .....	36
(4) Git、Github .....	36

## 1. セットアップ

こちらのページを参考にセットアップして下さい。

<https://github.com/mmochizuki/robocup2d/wiki/%E3%82%BB%E3%83%83%E3%83%88%E3%82%A2%E3%83%83%E3%83%97>

## 2. agent2d の使い方

### (1) 参考資料

RoboCup サッカー2D シミュレーションリーグ解説：仕組みと環境構築

[https://www.jstage.jst.go.jp/article/jsoft/23/5/23\\_714/pdf/-char/ja](https://www.jstage.jst.go.jp/article/jsoft/23/5/23_714/pdf/-char/ja)

RoboCup サッカー2D シミュレーションリーグ解説：サンプルエージェントを使ったチーム開発

[https://www.jstage.jst.go.jp/article/jsoft/23/6/23\\_838/pdf](https://www.jstage.jst.go.jp/article/jsoft/23/6/23_838/pdf)

ロボカップサッカーシミュレーション 2D リーグ必勝ガイド

<https://jaist.dl.osdn.jp/rctools/46021/RoboCup2DGuideBook-1.0.pdf>

RoboCup サッカー2D シミュレーション講習会@秋キャンプ 2011

下記のページの「講習会」と記載してある部分の「occersim2d-slide.pdf」というファイルをダウンロードして下さい。

<http://rc-oz.osdn.jp/pukiwiki/index.php?cmd=read&page=Event%2F2011%2FCamp&word=2011>

サッカーシミュレーションリーグ・情報処理学会電子図書館

[https://ipsj.ixsq.nii.ac.jp/ej/index.php?action=pages\\_view\\_main&active\\_action=repository\\_action\\_common\\_download&item\\_id=70557&item\\_no=1&attribute\\_id=1&file\\_no=1&page\\_id=13&block\\_id=8](https://ipsj.ixsq.nii.ac.jp/ej/index.php?action=pages_view_main&active_action=repository_action_common_download&item_id=70557&item_no=1&attribute_id=1&file_no=1&page_id=13&block_id=8)

## (2) librcsc(ライブラリ)

agent2d は librcsc というライブラリを使ってプログラムされています。librcsc は boost を使用して作られています。librcsc については、ロボカップサッカーシミュレーション 2D リーグ必勝ガイドの「第3章チーム開発 (p 39~)」をご覧ください。

<https://jaist.dl.osdn.jp/rctools/46021/RoboCup2DGuideBook-1.0.pdf>

また、必勝ガイドの p 46 から解説されている「Doxygen によるリファレンス生成」によってリファレンスを作成しました。下記のページの「librcsc\_index.zip」をダウンロードして、フォルダの中にある「index.html」ファイルをクリックすると、ライブラリの構造が分かります。

<https://github.com/mmochizuki/robocup2d>

## (3) フォルダ構造

agent2d は C++で書かれているため、実行するにはコンパイルが必要となります。コンパイル前の agent2d は、ファイルとフォルダを合わせて 234 個、容量は 1.9MB です。コンパイル後の agent2d は、ファイルとフォルダを合計で 332 個、容量は 86.8MB まで増えています。プログラムを修正する時には、自分でコンパイルが必要となるため、フォルダ構造とコンパイルの過程について説明します。

### ①コンパイル前

コンパイル前の agent2d は、ファイルとフォルダを合わせると 234 個、容量は 1.9MB です。src フォルダの中にある、「bhv\_basic\_move.cpp」、「bhv\_basic\_tackle.cpp」、「bhv\_normal\_dribble.cpp」などのファイルの中に、librcsc のライブラリを使って一つ一つの動作を作り込んでいます。これらのファイルのパラメーターを変化させれば、動きを変化させることができます。

— AUTHORS

— COPYING

— ChangeLog

— INSTALL

— Makefile.am

— Makefile.in

— NEWS

— NEWS.en

— README

— aclocal.m4

- |—— bootstrap
- |—— config
  - | |—— depcomp
  - | |—— install-sh
  - | |—— missing
- |—— config.h.in
- |—— configure
- |—— configure.ac
- |—— m4
  - | |—— ax\_boost\_base.m4
- |—— src
  - |—— Makefile.am
  - |—— Makefile.in
  - |—— bhv\_basic\_move.cpp
  - |—— bhv\_basic\_move.h
  - |—— bhv\_basic\_offensive\_kick.cpp
  - |—— bhv\_basic\_offensive\_kick.h
  - |—— bhv\_basic\_tackle.cpp
  - |—— bhv\_basic\_tackle.h
  - |—— bhv\_custom\_before\_kick\_off.cpp
  - |—— bhv\_custom\_before\_kick\_off.h
  - |—— bhv\_go\_to\_static\_ball.cpp
  - |—— bhv\_go\_to\_static\_ball.h
  - |—— bhv\_goalie\_basic\_move.cpp
  - |—— bhv\_goalie\_basic\_move.h
  - |—— bhv\_goalie\_chase\_ball.cpp
  - |—— bhv\_goalie\_chase\_ball.h
  - |—— bhv\_goalie\_free\_kick.cpp
  - |—— bhv\_goalie\_free\_kick.h
  - |—— bhv\_penalty\_kick.cpp
  - |—— bhv\_penalty\_kick.h
  - |—— bhv\_prepare\_set\_play\_kick.cpp
  - |—— bhv\_prepare\_set\_play\_kick.h
  - |—— bhv\_set\_play.cpp
  - |—— bhv\_set\_play.h
  - |—— bhv\_set\_play\_free\_kick.cpp

- |—— bhv\_set\_play\_free\_kick.h
- |—— bhv\_set\_play\_goal\_kick.cpp
- |—— bhv\_set\_play\_goal\_kick.h
- |—— bhv\_set\_play\_indirect\_free\_kick.cpp
- |—— bhv\_set\_play\_indirect\_free\_kick.h
- |—— bhv\_set\_play\_kick\_in.cpp
- |—— bhv\_set\_play\_kick\_in.h
- |—— bhv\_set\_play\_kick\_off.cpp
- |—— bhv\_set\_play\_kick\_off.h
- |—— bhv\_their\_goal\_kick\_move.cpp
- |—— bhv\_their\_goal\_kick\_move.h
- |—— chain\_action
  - | |—— actgen\_action\_chain\_length\_filter.h
  - | |—— actgen\_cross.cpp
  - | |—— actgen\_cross.h
  - | |—— actgen\_direct\_pass.cpp
  - | |—— actgen\_direct\_pass.h
  - | |—— actgen\_self\_pass.cpp
  - | |—— actgen\_self\_pass.h
  - | |—— actgen\_shoot.cpp
  - | |—— actgen\_shoot.h
  - | |—— actgen\_short\_dribble.cpp
  - | |—— actgen\_short\_dribble.h
  - | |—— actgen\_simple\_dribble.cpp
  - | |—— actgen\_simple\_dribble.h
  - | |—— actgen\_strict\_check\_pass.cpp
  - | |—— actgen\_strict\_check\_pass.h
  - | |—— action\_chain\_graph.cpp
  - | |—— action\_chain\_graph.h
  - | |—— action\_chain\_holder.cpp
  - | |—— action\_chain\_holder.h
  - | |—— action\_generator.h
  - | |—— action\_state\_pair.h
  - | |—— bhv\_chain\_action.cpp
  - | |—— bhv\_chain\_action.h
  - | |—— bhv\_normal\_dribble.cpp

- | |—— bhv\_normal\_dribble.h
- | |—— bhv\_pass\_kick\_find\_receiver.cpp
- | |—— bhv\_pass\_kick\_find\_receiver.h
- | |—— bhv\_strict\_check\_shoot.cpp
- | |—— bhv\_strict\_check\_shoot.h
- | |—— body\_force\_shoot.cpp
- | |—— body\_force\_shoot.h
- | |—— clear\_ball.cpp
- | |—— clear\_ball.h
- | |—— clear\_generator.cpp
- | |—— clear\_generator.h
- | |—— cooperative\_action.cpp
- | |—— cooperative\_action.h
- | |—— cross\_generator.cpp
- | |—— cross\_generator.h
- | |—— dribble.cpp
- | |—— dribble.h
- | |—— field\_analyzer.cpp
- | |—— field\_analyzer.h
- | |—— field\_evaluator.h
- | |—— hold\_ball.cpp
- | |—— hold\_ball.h
- | |—— neck\_turn\_to\_receiver.cpp
- | |—— neck\_turn\_to\_receiver.h
- | |—— pass.cpp
- | |—— pass.h
- | |—— pass\_checker.h
- | |—— predict\_ball\_object.h
- | |—— predict\_player\_object.h
- | |—— predict\_state.cpp
- | |—— predict\_state.h
- | |—— self\_pass\_generator.cpp
- | |—— self\_pass\_generator.h
- | |—— shoot.cpp
- | |—— shoot.h
- | |—— shoot\_generator.cpp



- | |—— shoot\_generator.h
- | |—— short\_dribble\_generator.cpp
- | |—— short\_dribble\_generator.h
- | |—— simple\_pass\_checker.cpp
- | |—— simple\_pass\_checker.h
- | |—— strict\_check\_pass\_generator.cpp
- | |—— strict\_check\_pass\_generator.h
- | |—— tackle\_generator.cpp
- | |—— tackle\_generator.h
- |—— coach.conf
- |—— communication.h
- |—— formations-dt
  - | |—— before-kick-off.conf
  - | |—— defense-formation.conf
  - | |—— goal-kick-opp.conf
  - | |—— goal-kick-our.conf
  - | |—— goalie-catch-opp.conf
  - | |—— goalie-catch-our.conf
  - | |—— goalie-formation.conf
  - | |—— indirect-freekick-opp-formation.conf
  - | |—— indirect-freekick-our-formation.conf
  - | |—— kickin-our-formation.conf
  - | |—— normal-formation.conf
  - | |—— offense-formation.conf
  - | |—— setplay-opp-formation.conf
  - | |—— setplay-our-formation.conf
- |—— formations-keeper
  - | |—— before-kick-off.conf
  - | |—— defense-formation.conf
  - | |—— goal-kick-opp.conf
  - | |—— goal-kick-our.conf
  - | |—— goalie-catch-opp.conf
  - | |—— goalie-catch-our.conf
  - | |—— goalie-formation.conf
  - | |—— indirect-freekick-opp-formation.conf
  - | |—— indirect-freekick-our-formation.conf

- | |—— kickin-our-formation.conf
- | |—— normal-formation.conf
- | |—— offense-formation.conf
- | |—— setplay-opp-formation.conf
- | |—— setplay-our-formation.conf
- |—— formations-taker
  - | |—— before-kick-off.conf
  - | |—— defense-formation.conf
  - | |—— goal-kick-opp.conf
  - | |—— goal-kick-our.conf
  - | |—— goalie-catch-opp.conf
  - | |—— goalie-catch-our.conf
  - | |—— goalie-formation.conf
  - | |—— indirect-freekick-opp-formation.conf
  - | |—— indirect-freekick-our-formation.conf
  - | |—— kickin-our-formation.conf
  - | |—— normal-formation.conf
  - | |—— offense-formation.conf
  - | |—— setplay-opp-formation.conf
  - | |—— setplay-our-formation.conf
- |—— intention\_receive.cpp
- |—— intention\_receive.h
- |—— intention\_wait\_after\_set\_play\_kick.cpp
- |—— intention\_wait\_after\_set\_play\_kick.h
- |—— keepaway.sh.in
- |—— keepaway\_communication.cpp
- |—— keepaway\_communication.h
- |—— main\_coach.cpp
- |—— main\_player.cpp
- |—— main\_trainer.cpp
- |—— neck\_default\_intercept\_neck.cpp
- |—— neck\_default\_intercept\_neck.h
- |—— neck\_goalie\_turn\_neck.cpp
- |—— neck\_goalie\_turn\_neck.h
- |—— neck\_offensive\_intercept\_neck.cpp
- |—— neck\_offensive\_intercept\_neck.h

- |—— player.conf
- |—— role\_center\_back.cpp
- |—— role\_center\_back.h
- |—— role\_center\_forward.cpp
- |—— role\_center\_forward.h
- |—— role\_defensive\_half.cpp
- |—— role\_defensive\_half.h
- |—— role\_goalie.cpp
- |—— role\_goalie.h
- |—— role\_keepaway\_keeper.cpp
- |—— role\_keepaway\_keeper.h
- |—— role\_keepaway\_taker.cpp
- |—— role\_keepaway\_taker.h
- |—— role\_offensive\_half.cpp
- |—— role\_offensive\_half.h
- |—— role\_sample.cpp
- |—— role\_sample.h
- |—— role\_side\_back.cpp
- |—— role\_side\_back.h
- |—— role\_side\_forward.cpp
- |—— role\_side\_forward.h
- |—— role\_side\_half.cpp
- |—— role\_side\_half.h
- |—— sample\_coach.cpp
- |—— sample\_coach.h
- |—— sample\_communication.cpp
- |—— sample\_communication.h
- |—— sample\_field\_evaluator.cpp
- |—— sample\_field\_evaluator.h
- |—— sample\_player.cpp
- |—— sample\_player.h
- |—— sample\_trainer.cpp
- |—— sample\_trainer.h
- |—— soccer\_role.cpp
- |—— soccer\_role.h
- |—— start-debug.sh

- |—— start-offline.sh
- |—— start.sh.in
- |—— strategy.cpp
- |—— strategy.h
- |—— team\_logo.xpm
- |—— train.sh.in
- |—— view\_tactical.cpp
- |—— view\_tactical.h

## ②コンパイル後

コンパイル後の `agent2d` は、ファイルとフォルダを合わせると 332 個、容量は 86.8MB です。コンパイルによってファイルが 100 個くらい増えていますが、ほとんどは「`sample_player-actgen_cross.o`」といよように拡張子が「.o」の機械語のファイルです。その他に増えたのは、「`sample_player`」などの実行ファイル、`Makefile` などのビルドに必要なファイルです。

- AUTHORS
- |—— COPYING
- |—— ChangeLog
- |—— INSTALL
- |—— Makefile
- |—— Makefile.am
- |—— Makefile.in
- |—— NEWS
- |—— NEWS.en
- |—— README
- |—— aclocal.m4
- |—— autom4te.cache
  - | |—— output.0
  - | |—— output.1
  - | |—— requests
  - | |—— traces.0
  - | |—— traces.1
- |—— bootstrap
- |—— config

- | |—— compile
- | |—— depcomp
- | |—— install-sh
- | |—— missing
- |—— config.h
- |—— config.h.in
- |—— config.h.in~
- |—— config.log
- |—— config.status
- |—— configure
- |—— configure.ac
- |—— m4
  - |—— ax\_boost\_base.m4
- |—— src
  - |—— Makefile
  - |—— Makefile.am
  - |—— Makefile.in
  - |—— bhv\_basic\_move.cpp
  - |—— bhv\_basic\_move.h
  - |—— bhv\_basic\_offensive\_kick.cpp
  - |—— bhv\_basic\_offensive\_kick.h
  - |—— bhv\_basic\_tackle.cpp
  - |—— bhv\_basic\_tackle.h
  - |—— bhv\_custom\_before\_kick\_off.cpp
  - |—— bhv\_custom\_before\_kick\_off.h
  - |—— bhv\_go\_to\_static\_ball.cpp
  - |—— bhv\_go\_to\_static\_ball.h
  - |—— bhv\_goalie\_basic\_move.cpp
  - |—— bhv\_goalie\_basic\_move.h
  - |—— bhv\_goalie\_chase\_ball.cpp
  - |—— bhv\_goalie\_chase\_ball.h
  - |—— bhv\_goalie\_free\_kick.cpp
  - |—— bhv\_goalie\_free\_kick.h
  - |—— bhv\_penalty\_kick.cpp
  - |—— bhv\_penalty\_kick.h
  - |—— bhv\_prepare\_set\_play\_kick.cpp

- | |—— bhv\_prepare\_set\_play\_kick.h
- | |—— bhv\_set\_play.cpp
- | |—— bhv\_set\_play.h
- | |—— bhv\_set\_play\_free\_kick.cpp
- | |—— bhv\_set\_play\_free\_kick.h
- | |—— bhv\_set\_play\_goal\_kick.cpp
- | |—— bhv\_set\_play\_goal\_kick.h
- | |—— bhv\_set\_play\_indirect\_free\_kick.cpp
- | |—— bhv\_set\_play\_indirect\_free\_kick.h
- | |—— bhv\_set\_play\_kick\_in.cpp
- | |—— bhv\_set\_play\_kick\_in.h
- | |—— bhv\_set\_play\_kick\_off.cpp
- | |—— bhv\_set\_play\_kick\_off.h
- | |—— bhv\_their\_goal\_kick\_move.cpp
- | |—— bhv\_their\_goal\_kick\_move.h
- | |—— chain\_action
  - | | |—— actgen\_action\_chain\_length\_filter.h
  - | | |—— actgen\_cross.cpp
  - | | |—— actgen\_cross.h
  - | | |—— actgen\_direct\_pass.cpp
  - | | |—— actgen\_direct\_pass.h
  - | | |—— actgen\_self\_pass.cpp
  - | | |—— actgen\_self\_pass.h
  - | | |—— actgen\_shoot.cpp
  - | | |—— actgen\_shoot.h
  - | | |—— actgen\_short\_dribble.cpp
  - | | |—— actgen\_short\_dribble.h
  - | | |—— actgen\_simple\_dribble.cpp
  - | | |—— actgen\_simple\_dribble.h
  - | | |—— actgen\_strict\_check\_pass.cpp
  - | | |—— actgen\_strict\_check\_pass.h
  - | | |—— action\_chain\_graph.cpp
  - | | |—— action\_chain\_graph.h
  - | | |—— action\_chain\_holder.cpp
  - | | |—— action\_chain\_holder.h
  - | | |—— action\_generator.h

			——	action_state_pair.h
			——	bhv_chain_action.cpp
			——	bhv_chain_action.h
			——	bhv_normal_dribble.cpp
			——	bhv_normal_dribble.h
			——	bhv_pass_kick_find_receiver.cpp
			——	bhv_pass_kick_find_receiver.h
			——	bhv_strict_check_shoot.cpp
			——	bhv_strict_check_shoot.h
			——	body_force_shoot.cpp
			——	body_force_shoot.h
			——	clear_ball.cpp
			——	clear_ball.h
			——	clear_generator.cpp
			——	clear_generator.h
			——	cooperative_action.cpp
			——	cooperative_action.h
			——	cross_generator.cpp
			——	cross_generator.h
			——	dribble.cpp
			——	dribble.h
			——	field_analyzer.cpp
			——	field_analyzer.h
			——	field_evaluator.h
			——	hold_ball.cpp
			——	hold_ball.h
			——	neck_turn_to_receiver.cpp
			——	neck_turn_to_receiver.h
			——	pass.cpp
			——	pass.h
			——	pass_checker.h
			——	predict_ball_object.h
			——	predict_player_object.h
			——	predict_state.cpp
			——	predict_state.h
			——	self_pass_generator.cpp

- | | |—— self\_pass\_generator.h
- | | |—— shoot.cpp
- | | |—— shoot.h
- | | |—— shoot\_generator.cpp
- | | |—— shoot\_generator.h
- | | |—— short\_dribble\_generator.cpp
- | | |—— short\_dribble\_generator.h
- | | |—— simple\_pass\_checker.cpp
- | | |—— simple\_pass\_checker.h
- | | |—— strict\_check\_pass\_generator.cpp
- | | |—— strict\_check\_pass\_generator.h
- | | |—— tackle\_generator.cpp
- | | |—— tackle\_generator.h
- | |—— coach.conf
- | |—— communication.h
- | |—— formations-dt
  - | | |—— before-kick-off.conf
  - | | |—— defense-formation.conf
  - | | |—— defense-formation\_1.conf
  - | | |—— goal-kick-opp.conf
  - | | |—— goal-kick-our.conf
  - | | |—— goalie-catch-opp.conf
  - | | |—— goalie-catch-our.conf
  - | | |—— goalie-formation.conf
  - | | |—— indirect-freekick-opp-formation.conf
  - | | |—— indirect-freekick-our-formation.conf
  - | | |—— kickin-our-formation.conf
  - | | |—— normal-formation.conf
  - | | |—— offense-formation.conf
  - | | |—— setplay-opp-formation.conf
  - | | |—— setplay-our-formation.conf
- | |—— formations-keeper
  - | | |—— before-kick-off.conf
  - | | |—— defense-formation.conf
  - | | |—— goal-kick-opp.conf
  - | | |—— goal-kick-our.conf



- | | |—— goalie-catch-opp.conf
- | | |—— goalie-catch-our.conf
- | | |—— goalie-formation.conf
- | | |—— indirect-freekick-opp-formation.conf
- | | |—— indirect-freekick-our-formation.conf
- | | |—— kickin-our-formation.conf
- | | |—— normal-formation.conf
- | | |—— offense-formation.conf
- | | |—— setplay-opp-formation.conf
- | | |—— setplay-our-formation.conf
- | |—— formations-taker
- | | |—— before-kick-off.conf
- | | |—— defense-formation.conf
- | | |—— goal-kick-opp.conf
- | | |—— goal-kick-our.conf
- | | |—— goalie-catch-opp.conf
- | | |—— goalie-catch-our.conf
- | | |—— goalie-formation.conf
- | | |—— indirect-freekick-opp-formation.conf
- | | |—— indirect-freekick-our-formation.conf
- | | |—— kickin-our-formation.conf
- | | |—— normal-formation.conf
- | | |—— offense-formation.conf
- | | |—— setplay-opp-formation.conf
- | | |—— setplay-our-formation.conf
- | |—— intention\_receive.cpp
- | |—— intention\_receive.h
- | |—— intention\_wait\_after\_set\_play\_kick.cpp
- | |—— intention\_wait\_after\_set\_play\_kick.h
- | |—— keepaway.sh
- | |—— keepaway.sh.in
- | |—— keepaway\_communication.cpp
- | |—— keepaway\_communication.h
- | |—— main\_coach.cpp
- | |—— main\_player.cpp
- | |—— main\_trainer.cpp

- |   |—— neck\_default\_intercept\_neck.cpp
- |   |—— neck\_default\_intercept\_neck.h
- |   |—— neck\_goalie\_turn\_neck.cpp
- |   |—— neck\_goalie\_turn\_neck.h
- |   |—— neck\_offensive\_intercept\_neck.cpp
- |   |—— neck\_offensive\_intercept\_neck.h
- |   |—— player.conf
- |   |—— role\_center\_back.cpp
- |   |—— role\_center\_back.h
- |   |—— role\_center\_forward.cpp
- |   |—— role\_center\_forward.h
- |   |—— role\_defensive\_half.cpp
- |   |—— role\_defensive\_half.h
- |   |—— role\_goalie.cpp
- |   |—— role\_goalie.h
- |   |—— role\_keepaway\_keeper.cpp
- |   |—— role\_keepaway\_keeper.h
- |   |—— role\_keepaway\_taker.cpp
- |   |—— role\_keepaway\_taker.h
- |   |—— role\_offensive\_half.cpp
- |   |—— role\_offensive\_half.h
- |   |—— role\_sample.cpp
- |   |—— role\_sample.h
- |   |—— role\_side\_back.cpp
- |   |—— role\_side\_back.h
- |   |—— role\_side\_forward.cpp
- |   |—— role\_side\_forward.h
- |   |—— role\_side\_half.cpp
- |   |—— role\_side\_half.h
- |   |—— sample\_coach
- |   |—— sample\_coach-main\_coach.o
- |   |—— sample\_coach-sample\_coach.o
- |   |—— sample\_coach.cpp
- |   |—— sample\_coach.h
- |   |—— sample\_communication.cpp
- |   |—— sample\_communication.h

- |   |—— sample\_field\_evaluator.cpp
- |   |—— sample\_field\_evaluator.h
- |   |—— sample\_player
- |   |—— sample\_player-actgen\_cross.o
- |   |—— sample\_player-actgen\_direct\_pass.o
- |   |—— sample\_player-actgen\_self\_pass.o
- |   |—— sample\_player-actgen\_shoot.o
- |   |—— sample\_player-actgen\_short\_dribble.o
- |   |—— sample\_player-actgen\_simple\_dribble.o
- |   |—— sample\_player-actgen\_strict\_check\_pass.o
- |   |—— sample\_player-action\_chain\_graph.o
- |   |—— sample\_player-action\_chain\_holder.o
- |   |—— sample\_player-bhv\_basic\_move.o
- |   |—— sample\_player-bhv\_basic\_offensive\_kick.o
- |   |—— sample\_player-bhv\_basic\_tackle.o
- |   |—— sample\_player-bhv\_chain\_action.o
- |   |—— sample\_player-bhv\_custom\_before\_kick\_off.o
- |   |—— sample\_player-bhv\_go\_to\_static\_ball.o
- |   |—— sample\_player-bhv\_goalie\_basic\_move.o
- |   |—— sample\_player-bhv\_goalie\_chase\_ball.o
- |   |—— sample\_player-bhv\_goalie\_free\_kick.o
- |   |—— sample\_player-bhv\_normal\_dribble.o
- |   |—— sample\_player-bhv\_pass\_kick\_find\_receiver.o
- |   |—— sample\_player-bhv\_penalty\_kick.o
- |   |—— sample\_player-bhv\_prepare\_set\_play\_kick.o
- |   |—— sample\_player-bhv\_set\_play.o
- |   |—— sample\_player-bhv\_set\_play\_free\_kick.o
- |   |—— sample\_player-bhv\_set\_play\_goal\_kick.o
- |   |—— sample\_player-bhv\_set\_play\_indirect\_free\_kick.o
- |   |—— sample\_player-bhv\_set\_play\_kick\_in.o
- |   |—— sample\_player-bhv\_set\_play\_kick\_off.o
- |   |—— sample\_player-bhv\_strict\_check\_shoot.o
- |   |—— sample\_player-bhv\_their\_goal\_kick\_move.o
- |   |—— sample\_player-body\_force\_shoot.o
- |   |—— sample\_player-clear\_ball.o
- |   |—— sample\_player-clear\_generator.o

- |   |—— sample\_player-cooperative\_action.o
- |   |—— sample\_player-cross\_generator.o
- |   |—— sample\_player-dribble.o
- |   |—— sample\_player-field\_analyzer.o
- |   |—— sample\_player-hold\_ball.o
- |   |—— sample\_player-intention\_receive.o
- |   |—— sample\_player-intention\_wait\_after\_set\_play\_kick.o
- |   |—— sample\_player-keepaway\_communication.o
- |   |—— sample\_player-main\_player.o
- |   |—— sample\_player-neck\_default\_intercept\_neck.o
- |   |—— sample\_player-neck\_goalie\_turn\_neck.o
- |   |—— sample\_player-neck\_offensive\_intercept\_neck.o
- |   |—— sample\_player-neck\_turn\_to\_receiver.o
- |   |—— sample\_player-pass.o
- |   |—— sample\_player-predict\_state.o
- |   |—— sample\_player-role\_center\_back.o
- |   |—— sample\_player-role\_center\_forward.o
- |   |—— sample\_player-role\_defensive\_half.o
- |   |—— sample\_player-role\_goalie.o
- |   |—— sample\_player-role\_keepaway\_keeper.o
- |   |—— sample\_player-role\_keepaway\_taker.o
- |   |—— sample\_player-role\_offensive\_half.o
- |   |—— sample\_player-role\_sample.o
- |   |—— sample\_player-role\_side\_back.o
- |   |—— sample\_player-role\_side\_forward.o
- |   |—— sample\_player-role\_side\_half.o
- |   |—— sample\_player-sample\_communication.o
- |   |—— sample\_player-sample\_field\_evaluator.o
- |   |—— sample\_player-sample\_player.o
- |   |—— sample\_player-self\_pass\_generator.o
- |   |—— sample\_player-shoot.o
- |   |—— sample\_player-shoot\_generator.o
- |   |—— sample\_player-short\_dribble\_generator.o
- |   |—— sample\_player-simple\_pass\_checker.o
- |   |—— sample\_player-soccer\_role.o
- |   |—— sample\_player-strategy.o

```

|   |—— sample_player-strict_check_pass_generator.o
|   |—— sample_player-tackle_generator.o
|   |—— sample_player-view_tactical.o
|   |—— sample_player.cpp
|   |—— sample_player.h
|   |—— sample_trainer
|   |—— sample_trainer-main_trainer.o
|   |—— sample_trainer-sample_trainer.o
|   |—— sample_trainer.cpp
|   |—— sample_trainer.h
|   |—— soccer_role.cpp
|   |—— soccer_role.h
|   |—— start-debug.sh
|   |—— start-offline.sh
|   |—— start.sh
|   |—— start.sh.in
|   |—— strategy.cpp
|   |—— strategy.h
|   |—— team_logo.xpm
|   |—— train.sh
|   |—— train.sh.in
|   |—— view_tactical.cpp
|   |—— view_tactical.h
|—— stamp-h1
—— start.sh

```

## (4) コンパイル

### ①コンパイル (ビルドの過程)

C 言語のソースファイルから実行ファイルを作ることをコンパイルと言います。コンパイルは、「プリプロセス」、「コンパイル」、「アセンブル」、「リンク」の4つの過程に分けることができます。詳しくは、下記のページをご覧ください。ちなみに、紹介したページの中にある「gcc」を「g++」にすれば、C++をコンパイルすることができます。

C 言語がコンパイルされて実行可能になるまでの流れ

<http://aoking.hatenablog.jp/entry/20121109/1352457273>

「ビルド」という作業は何を指しているのか

<https://www.atmarkit.co.jp/ait/articles/1105/23/news128.html>

ダイナミックリンクとスタティックリンク

<https://www.atmarkit.co.jp/ait/articles/1105/27/news111.html>

もっと詳しく知りたい方は、「C 言語本格入門～基礎知識からコンピュータの本質まで」の「1-4 C 言語の開発者ツールの役割」をご覧ください。

<http://gihyo.jp/book/2018/978-4-7741-9616-9>

### ②Make

ファイル数が少ない時はコマンドラインを使ってコンパイルをしても問題ありませんが、ファイル数が増えてくる何度もコマンドを打たなければならず大変です。そこで、**Makefile**を作成して **make** を使うと一度でコンパイルを実行することができます。

make コマンドを使ってみよう

<https://www.miraclelinux.com/tech-blog/0icygs>

make を使ってソフトウェアをビルドしてみよう

<https://www.atmarkit.co.jp/ait/articles/1106/07/news131.html>

Makefile をいろいろ書き換えながらビルドしてみよう

<https://www.atmarkit.co.jp/ait/articles/1106/10/news115.html>

### ③GNU Autotool

ファイル数が増えた場合や、自分以外の環境でビルドする時には「GNU Autotool」を使って **Makefile** を作成すると便利で、Agent2d も「GNU Autotool」を使って、ソースファイルをビルドしています。Autotool について知りたい方は下記のサイトをご覧ください。

Autotools

<https://ja.wikipedia.org/wiki/Autotools>

autotools を使ってみよう

<https://www.miraclelinux.com/tech-blog/reqys8>

GNU Autotools で「Hello, World」

<https://qiita.com/narupo/items/f63b8e768f17ce50f398>

Autotools ( automake, autoconf, libtool ) 使い方まとめ

<http://tamaobject.hatenablog.com/entry/2013/08/01/165119>

#### ④agent2d でファイルを追加した時

ドキュメント/ファイルの追加・削除・リネーム

<http://rctools.osdn.jp/pukiwiki/index.php?%A5%C9%A5%AD%A5%E5%A5%E1%A5%F3%A5%C8/%A5%D5%A5%A1%A5%A4%A5%EB%A4%CE%C4%C9%B2%C3%A1%A6%B%A%EF%BD%FC%A1%A6%A5%EA%A5%CD%A1%BC%A5%E0>

#### ⑤agent2d のライブラリをプログラムに同梱する方法

agent2d/公開用バイナリ作成方法

<http://rctools.osdn.jp/pukiwiki/index.php?agent2d/%B8%F8%B3%AB%CD%D1%A5%D0%A5%A4%A5%CA%A5%EA%BA%EE%C0%AE%CA%FD%CB%A1>

公開用パッケージの作り方

<http://rctools.osdn.jp/pukiwiki/index.php?cmd=read&page=agent2d%2F%B8%F8%B3%AB%CD%D1%A5%D1%A5%C3%A5%B1%A1%BC%A5%B8%A4%CE%BA%EE%A4%EA%CA%FD>

## (5) プログラムの開始までの流れ (start.sh)

各自が作成したメインフォルダの中にある「start.sh」→ src フォルダの中にある「start.sh」という流れで動いていきます。src フォルダの中にある「start.sh」はシェルスクリプトで書かれており、主な流れは下記の通りです。

### ①LIBPATH の書き込み (12 行~20 行)

### ②実行ファイル、conf ファイル、port 番号などを変数に入れる (24 行~53 行)

下記のように変数にファイルへのパスや数字などをいれていきます。

```
player="${DIR}/sample_player"
coach="${DIR}/sample_coach"
teamname="HELIOS_base"
host="localhost"
port=6000
```

### ・ 検討事項 (2019/4/7)

「sample\_player」や「sample\_coach」は実行ファイルです。この実行ファイルを player という変数に入れているのは分かるのですが、プログラムのどの部分で実行しているかが分かったら教えて下さい。

### ③ヘルプの内容表示 (55 行~80 行)

### ④引数の処理 (82 行~262 行)

### ⑤その他の処理 (264 行~)

### ⑥start.sh を実行した後にコンソールに表示される内容

最後の部分に表示された内容を見ると、割り当てられた 18 名の選手の中から 11 名の選手を選択し、適切なポジションに割り当てていることが分かります。

\*\*\*\*\*

HELIOS base

Created by Hidehisa Akiyama and Hiroki Shimora

Copyright 2000-2007. Hidehisa Akiyama



Copyright 2007-2012. Hidehisa Akiyama and Hiroki Shimora  
All rights reserved.

\*\*\*\*\*

PING localhost (127.0.0.1) 56(84) bytes of data.

64 bytes from localhost (127.0.0.1): icmp\_seq=1 ttl=64 time=0.037 ms

--- localhost ping statistics ---

1 packets transmitted, 1 received, 0% packet loss, time 0ms

rtt min/avg/max/mdev = 0.037/0.037/0.037/0.000 ms

\*\*\*\*\*

librcsc 4.1.0

Copyright 2000 - 2007. Hidehisa Akiyama.

Copyright 2007 - 2011. Hidehisa Akiyama and Hiroki Shimora

All rights reserved.

\*\*\*\*\*

\*\*\*\*\*

This program is based on agent2d created by Hidehisa Akiyama.

Copyright 2006 - 2011. Hidehisa Akiyama and Hiroki Shimora.

All rights reserved.

\*\*\*\*\*

mochizuki2d: init ok. unum: 1 side: 1

mochizuki2d 1: KickTable created.

mochizuki2d 1: [-1, 0] set synch see mode.

\*\*\*\*\*

librcsc 4.1.0

Copyright 2000 - 2007. Hidehisa Akiyama.

Copyright 2007 - 2011. Hidehisa Akiyama and Hiroki Shimora

All rights reserved.

\*\*\*\*\*

\*\*\*\*\*

librcsc 4.1.0

Copyright 2000 - 2007. Hidehisa Akiyama.

Copyright 2007 - 2011. Hidehisa Akiyama and Hiroki Shimora

All rights reserved.

\*\*\*\*\*

\*\*\*\*\*

librcsc 4.1.0

Copyright 2000 - 2007. Hidehisa Akiyama.

Copyright 2007 - 2011. Hidehisa Akiyama and Hiroki Shimora

All rights reserved.

\*\*\*\*\*

\*\*\*\*\*

This program is based on agent2d created by Hidehisa Akiyama.

Copyright 2006 - 2011. Hidehisa Akiyama and Hiroki Shimora.

All rights reserved.

\*\*\*\*\*

\*\*\*\*\*

librcsc 4.1.0

Copyright 2000 - 2007. Hidehisa Akiyama.

Copyright 2007 - 2011. Hidehisa Akiyama and Hiroki Shimora

All rights reserved.

\*\*\*\*\*

mochizuki2d: init ok. unum: 2 side: 1

\*\*\*\*\*

This program is based on agent2d created by Hidehisa Akiyama.

Copyright 2006 - 2011. Hidehisa Akiyama and Hiroki Shimora.

All rights reserved.

\*\*\*\*\*

mochizuki2d: init ok. unum: 3 side: 1

\*\*\*\*\*

This program is based on agent2d created by Hidehisa Akiyama.

Copyright 2006 - 2011. Hidehisa Akiyama and Hiroki Shimora.

All rights reserved.

\*\*\*\*\*

mochizuki2d: init ok. unum: 4 side: 1

\*\*\*\*\*

This program is based on agent2d created by Hidehisa Akiyama.

Copyright 2006 - 2011. Hidehisa Akiyama and Hiroki Shimora.

All rights reserved.

\*\*\*\*\*

mochizuki2d: init ok. unum: 5 side: 1

mochizuki2d 3: KickTable created.

mochizuki2d 3: [-1, 0] set synch see mode.

\*\*\*\*\*

librcsc 4.1.0

Copyright 2000 - 2007. Hidehisa Akiyama.

Copyright 2007 - 2011. Hidehisa Akiyama and Hiroki Shimora

All rights reserved.

\*\*\*\*\*

mochizuki2d 2: KickTable created.

mochizuki2d 2: [-1, 0] set synch see mode.

\*\*\*\*\*

librcsc 4.1.0

Copyright 2000 - 2007. Hidehisa Akiyama.

Copyright 2007 - 2011. Hidehisa Akiyama and Hiroki Shimora

All rights reserved.

\*\*\*\*\*

mochizuki2d 5: KickTable created.

mochizuki2d 5: [0, 1] set synch see mode.

\*\*\*\*\*

This program is based on agent2d created by Hidehisa Akiyama.

Copyright 2006 - 2011. Hidehisa Akiyama and Hiroki Shimora.

All rights reserved.

\*\*\*\*\*

\*\*\*\*\*

librcsc 4.1.0

Copyright 2000 - 2007. Hidehisa Akiyama.

Copyright 2007 - 2011. Hidehisa Akiyama and Hiroki Shimora

All rights reserved.

\*\*\*\*\*

mochizuki2d: init ok. unum: 6 side: 1

\*\*\*\*\*

This program is based on agent2d created by Hidehisa Akiyama.

Copyright 2006 - 2011. Hidehisa Akiyama and Hiroki Shimora.

All rights reserved.

\*\*\*\*\*

mochizuki2d: init ok. unum: 7 side: 1

\*\*\*\*\*

This program is based on agent2d created by Hidehisa Akiyama.  
Copyright 2006 - 2011. Hidehisa Akiyama and Hiroki Shimora.  
All rights reserved.

\*\*\*\*\*

mochizuki2d: init ok. unum: 8 side: 1

\*\*\*\*\*

librcsc 4.1.0

Copyright 2000 - 2007. Hidehisa Akiyama.

Copyright 2007 - 2011. Hidehisa Akiyama and Hiroki Shimora

All rights reserved.

\*\*\*\*\*

mochizuki2d 4: KickTable created.

mochizuki2d 4: [-1, 0] set synch see mode.

mochizuki2d 4: [0, 2] missed last action?(1) last decision=[-1, 0]

\*\*\*\*\*

librcsc 4.1.0

Copyright 2000 - 2007. Hidehisa Akiyama.

Copyright 2007 - 2011. Hidehisa Akiyama and Hiroki Shimora

All rights reserved.

\*\*\*\*\*

\*\*\*\*\*

This program is based on agent2d created by Hidehisa Akiyama.

Copyright 2006 - 2011. Hidehisa Akiyama and Hiroki Shimora.

All rights reserved.

\*\*\*\*\*

mochizuki2d 7: KickTable created.

mochizuki2d 7: [-1, 0] set synch see mode.

mochizuki2d: init ok. unum: 9 side: 1

\*\*\*\*\*

librcsc 4.1.0

Copyright 2000 - 2007. Hidehisa Akiyama.

Copyright 2007 - 2011. Hidehisa Akiyama and Hiroki Shimora

All rights reserved.

\*\*\*\*\*

mochizuki2d 8: KickTable created.

mochizuki2d 6: KickTable created.

mochizuki2d 8: [0, 1]mochizuki2d 6: [-1, 0] set synch see mode.  
set synch see mode.

\*\*\*\*\*

This program is based on agent2d created by Hidehisa Akiyama.  
Copyright 2006 - 2011. Hidehisa Akiyama and Hiroki Shimora.  
All rights reserved.

\*\*\*\*\*

mochizuki2d: init ok. unum: 10 side: 1  
exit good bye  
mm@mm-

VirtualBox:~/t1\$ \*\*\*\*\*

This program is based on agent2d created by Hidehisa Akiyama.  
Copyright 2006 - 2011. Hidehisa Akiyama and Hiroki Shimora.  
All rights reserved.

\*\*\*\*\*

\*\*\*\*\*

librcsc 4.1.0  
Copyright 2000 - 2007. Hidehisa Akiyama.  
Copyright 2007 - 2011. Hidehisa Akiyama and Hiroki Shimora  
All rights reserved.

\*\*\*\*\*

\*\*\*\*\*

This program is based on agent2d created by Hidehisa Akiyama.  
Copyright 2006 - 2011. Hidehisa Akiyama and Hiroki Shimora.  
All rights reserved.

\*\*\*\*\*

mochizuki2d: init ok. unum: 11 side: 1  
mochizuki2d 9: KickTable created.  
mochizuki2d 9: [-1, 0] set synch see mode.  
mochizuki2d coach: [-1, 0] recv (ok eye on)  
mochizuki2d 10: KickTable created.  
mochizuki2d 10: [-1, 0] set synch see mode.  
mochizuki2d 11: KickTable created.  
mochizuki2d 11: [-1, 0] set synch see mode.  
id speed step inc power stam karea  
0 1.000 6 45.0 100.0 55.0 1.085

1	0.790	5	47.4	100.0	52.6	1.044
2	0.837	6	51.3	100.0	48.7	1.172
3	0.833	4	43.4	100.0	56.6	1.074
4	0.929	5	45.8	100.0	54.2	1.088
5	0.924	6	44.6	100.0	55.4	1.135
6	0.947	5	43.0	100.0	57.0	1.153
7	0.907	4	41.4	100.0	58.6	0.994
8	0.787	5	45.3	100.0	54.7	1.105
9	0.801	6	49.1	100.0	50.9	1.126
10	0.836	6	48.5	100.0	51.5	1.035
11	0.813	4	42.4	100.0	57.6	1.076
12	0.778	5	50.5	100.0	49.5	1.024
13	0.833	5	41.9	100.0	58.1	0.986
14	0.959	6	44.9	100.0	55.1	1.084
15	0.972	7	45.6	100.0	54.4	1.022
16	0.882	5	43.0	100.0	57.0	1.048
17	0.945	6	47.3	100.0	52.7	1.034

mochizuki2d coach: change player 1 to type 0

mochizuki2d coach: change player 11 to type 15

mochizuki2d coach: change player 2 to type 14

mochizuki2d coach: change player 3 to type 6

mochizuki2d coach: change player 10 to type 17

mochizuki2d coach: change player 9 to type 4

mochizuki2d coach: change player 6 to type 5

mochizuki2d coach: change player 4 to type 7

mochizuki2d coach: change player 5 to type 16

mochizuki2d coach: change player 7 to type 3

mochizuki2d coach: change player 8 to type 13

## (6) 修正を必要とするファイル

### ①フォーメーションの変更

src フォルダの中の「formations-dt」フォルダの中のファイル。詳細については、「3.フォーメーションの変更」を参照して下さい。

(図 1) formations-dt フォルダ構造

```
|  └── formations-dt
|    ├── before-kick-off.conf
|    ├── defense-formation.conf
|    ├── goal-kick-opp.conf
|    ├── goal-kick-our.conf
|    ├── goalie-catch-opp.conf
|    ├── goalie-catch-our.conf
|    ├── goalie-formation.conf
|    ├── indirect-freekick-opp-formation.conf
|    ├── indirect-freekick-our-formation.conf
|    ├── kickin-our-formation.conf
|    ├── normal-formation.conf
|    ├── offense-formation.conf
|    ├── setplay-opp-formation.conf
|    └── setplay-our-formation.conf
```

### ②ボールキックの行動評価関数

src フォルダの中の「sample\_field\_evaluator.cpp」に記載されており、以下のヘッダファイルを読み込んでいます。下記に記載しているヘッダファイルは、agent2d 中のファイルに限定しており、量が多くなってしまうため libresc のヘッダファイルは記載していません。

```
#include "sample_field_evaluator.h"
#include "field_analyzer.h"
#include "simple_pass_checker.h"
```

ちなみに、C++のプログラムでは「.cpp」ファイルを他のプログラムファイルから使用するためなどの情報を「.h」ファイルに記載することが多く、「.h」と対になった「.cpp」ファイルがあります。「field\_analyzer.h」、「field\_analyzer.cpp」、

“simple\_pass\_checker.h”、“simple\_pass\_checker.cpp”は **src** フォルダの中の「chain\_action」フォルダの中にあります。

### ③ポジショニング動作

**src** フォルダの中の「bhv\_basic\_move.cpp」に記載されています。以下のヘッダファイルを読み込んでいます。これらのファイルは **src** フォルダの中にあります。

```
#include "bhv_basic_move.h"
#include "strategy.h"
#include "bhv_basic_tackle.h"
```

### ④戦略（ポジショニング）

**src** フォルダの中の「strategy.cpp」に記載されています。このファイルの中でフォーメーションのポジションなどを規定しています。このプログラムでは、**src** フォルダの中にある以下のヘッダファイルを読み込んでいます。

```
#include "strategy.h"
#include "soccer_role.h"
#include "role_sample.h"
#include "role_center_back.h"
#include "role_center_forward.h"
#include "role_defensive_half.h"
#include "role_goalie.h"
#include "role_offensive_half.h"
#include "role_side_back.h"
#include "role_side_forward.h"
#include "role_side_half.h"
#include "role_keepaway_keeper.h"
#include "role_keepaway_taker.h"
```



## ⑤コミュニケーションルール

src フォルダの中の「sample\_communication.cpp」に記載されています。このプログラムでは、src フォルダの中にある以下のヘッダファイルを読み込んでいます。読み込んでいるファイルが「strategy.h」なので、フォーメーションを作った後に、それにふさわしいコミュニケーションルールを作る必要があることが分かります。

```
#include "sample_communication.h"
#include "strategy.h"
```

## ⑥プレイヤーの役割

src フォルダの中の「soccer\_role.cpp」に記載されています。このファイルの中では、「soccer\_role.h」以外の他のポジションのファイルは読み込んでいないため、作り込めばポジションごとに動きを変えられることができると思います。このプログラムでは、src フォルダの中にある以下のヘッダファイルを読み込んでいます。

```
#include "soccer_role.h"
```

## ⑦全体のコントロール

src フォルダの中の「sample\_player.cpp」に記載されています。このプログラムでは、src フォルダの中にある以下のヘッダファイルを読み込んでいます。

```
#include "sample_player.h"
#include "strategy.h"
#include "field_analyzer.h"
#include "action_chain_holder.h"
#include "sample_field_evaluator.h"
#include "soccer_role.h"
#include "sample_communication.h"
#include "keepaway_communication.h"
#include "bhv_penalty_kick.h"
#include "bhv_set_play.h"
#include "bhv_set_play_kick_in.h"
#include "bhv_set_play_indirect_free_kick.h"
#include "bhv_custom_before_kick_off.h"
#include "bhv_strict_check_shoot.h"
#include "view_tactical.h"
#include "intention_receive.h"
```

### 3. フォーメーションの変更

#### (1) フォルダ構造

src フォルダの中に「formations-dt」というフォルダがあり、下記のような構造となっています (図1)。

(図1) formations-dt フォルダ構造

```
|  └── formations-dt
|    ├── before-kick-off.conf
|    ├── defense-formation.conf
|    ├── goal-kick-opp.conf
|    ├── goal-kick-our.conf
|    ├── goalie-catch-opp.conf
|    ├── goalie-catch-our.conf
|    ├── goalie-formation.conf
|    ├── indirect-freekick-opp-formation.conf
|    ├── indirect-freekick-our-formation.conf
|    ├── kickin-our-formation.conf
|    ├── normal-formation.conf
|    ├── offense-formation.conf
|    ├── setplay-opp-formation.conf
|    └── setplay-our-formation.conf
```

フォーメーションファイルには、テキストファイルを直接修正するものと、fedit2 (フォーメーション編集ツール) を使って修正するものがあります。

#### ・直接修正するファイル

ファイルの最初に「Formation Static」と記載されている以下のファイルです。before-kick-off.conf、goal-kick-opp.conf、goal-kick-our.conf、goalie-catch-opp.conf、goalie-catch-our.conf です。before-kick-off.conf ファイルを修正すると、キックオフ時のポジションを修正できます。

#### ・fedit2 を使って修正するファイル

ファイルの最初に「Formation DelaunayTriangulation 2」と記載されている、上記以外のファイルです。

fedit2 ダウンロードページ

<http://rctools.osdn.jp/pukiwiki/index.php?fedit2>

## (2) fedit2 の使い方

端末で「fedit2」と打つと起動できます。使い方については、「RoboCup サッカー2D シミュレーション講習会@秋キャンプ 2011」のスライド 36 以降に記載されています。スライドは下記のページの「講習会」と記載してある部分の「occersim2d-slide.pdf」というファイルをダウンロードして下さい。

<http://rc-oz.osdn.jp/pukiwiki/index.php?cmd=read&page=Event%2F2011%2FCamp&word=2011>

ちなみに、fedit2 を使って「defense-formation.conf」というファイルを開くと 115 ヶ所のボールの位置に対する選手の場所を設定できます。ボールの位置に対する選手の場所を修正した場合は、メニューバーの中にある「Replace」ボタンを押すと、記録されます。「Replace」ボタンを押すことを忘れてしまうと記録されませんので、ボールの位置が変わるたびに「Replace」ボタンを押して下さい。

## 検討事項 (2019/4/7)

ロボカップサッカーシミュレーション 2D リーグ必勝ガイド

<https://jaist.dl.osdn.jp/rctools/46021/RoboCup2DGuideBook-1.0.pdf>

本書の 144 ページ以降に「FormationEditor」の使い方が記載されています。

---

FormationEditor を実行するには、以下のように --editor-mode オプションを付けて soccerwindow2 を起動します。

```
$ soccerwindow2 --editor-mode
```

起動後、メニューから“New Formation”を選択すると、画面が図 3.4 のような状態になります。図 3.4 FormationEditor の実行画面ダイアログに表示されている役割の名前は、役割クラスで定義している名前に対応しています。必要に応じて変更してください。また、役割配分が望みのものと異なるのであれば、ダイアログを操作して変更してください。後は以下の手順を実行するだけです。

## 第 3 章 チーム開発

1. ボールを移動
2. プレイヤを移動
3. “Record” ボタンで訓練データ保存
4. “Train” ボタンで学習を実施
5. 訓練データ作成と学習を繰り返し実行

## 6. メニューから保存して終了

作成したフォーメーションのファイルは、サンプルチームのフォーメーションとして使用可能です。サンプルチームの Strategy クラスをそのまま使うなら、味方チームのキックイン、ボールが敵陣に存在する場合、ボールが自陣に存在する場合、の3種類のフォーメーションを作成することになります

---

この説明を読む限りでは、「train」というプロセスが必要になりそうです。こちらの説明は FormationEditor 1 なので、私たちが使用している FormationEditor 2 とは違うかもしれませんが、FormationEditor 2 にも「train」というボタンはありますが、このボタンを押すことによってどのような影響があるかはわかりません。このボタンを押すとどのようなことが起こるかが分かった方は教えて下さい。

## 参考情報

### (1) Agent2d 関連

秋山さんホームページ

<http://rctools.osdn.jp/pukiwiki/>

秋山さん論文紹介ページ

<http://resweb2.jhk.adm.fukuoka-u.ac.jp/FukuokaUnivHtml/info/5687/R110J.html>

The RoboCup Soccer Simulator

<https://rcsoccersim.github.io/>

### (2) 世界大会出場チームのアルゴリズム説明 (Team Description Paper)

[https://wrighteagle2d.github.io/robocup\\_tdp.html](https://wrighteagle2d.github.io/robocup_tdp.html)

### (3) C、C++関連

(概要)

C 言語

<https://ja.wikipedia.org/wiki/C%E8%A8%80%E8%AA%9E>

C++

<https://ja.wikipedia.org/wiki/C%2B%2B>

(参考書籍)

C 言語本格入門～基礎知識からコンピュータの本質まで

<http://gihyo.jp/book/2018/978-4-7741-9616-9>

スラスラわかる C++ 第2版

<https://www.shoeisha.co.jp/book/detail/9784798153872>

### (4) Git、Github

Git/GitHub レベル別オススメ学習サイトまとめ完全保存版【2019.03】

<https://qiita.com/think-a-lot/items/b3c2e9060f46f5d4ea46>

いつやるの？Git 入門

<https://www.slideshare.net/matsukaz/git-17499005>

今日からはじめる GitHub ～ 初心者が Git をインストールして、プルリクできるようになるまでを解説

<https://employment.en-japan.com/engineerhub/entry/2017/01/31/110000>